

UNIVERZITET SINGIDUNUM
DEPARTMAN ZA POSLEDIPLOMSKE STUDIJE

MASTER PROGRAM: SAVREMENE INFORMACIONE TEHNOLOGIJE

Predrag Mijajlović

HIBRIDNO ŠIFROVANJE FAJLOVA

- Master rad -

UNIVERZITET SINGIDUNUM
DEPARTMAN ZA POSLEDIPLOMSKE STUDIJE

MASTER PROGRAM: SAVREMENE INFORMACIONE TEHNOLOGIJE

HIBRIDNO ŠIFROVANJE FAJLOVA

- Master rad -

Mentor:
Prof dr Mladen Veinović

Student:
Predrag Mijajlović

Br. indeksa:
M 9467/2008

Sažetak

U ovom master radu predložen je, analiziran i realizovan jedan hibridni sistem za kriptografsku zaštitu fajlova. Sistem se zasniva na teorijskim principima šifrovanja i dešifrovanja, primeni kriptografskih algoritama i odgovarajućih kriptografskih ključeva. Kod asimetrične kriptografije primenjen je RSA algoritam, a posebna pažnja je posvećena računarskoj implementaciji velikih prostih brojeva. Za simetričan sistem šifrovanja odabran je RC4 algoritam i date su njegove teorijske osnove. Na kraju rada prikazan je realizovani softverski paket koji omogućava primenu predloženog hibridnog sistema za kriptografsku zaštitu fajlova. Softver je realizovan u okruženju Microsoft Visual Studio 6.0 IDE.

Ključne reči: Kriptografija, zaštita fajlova, simetrični i asimetrični šifarski sistemi, računarska implementacija

Abstract

In this master work is proposed, analyzed and realized a hybrid system for cryptographic protection of files. The system is based on theoretical principles of encryption and decryption, implementation of cryptographic algorithms and the corresponding cryptographic keys. In asymmetric cryptography an RSA algorithm is applied, and special attention was paid to computer implementation of large prime numbers. For symmetric encryption system an RC4 algorithm was chosen and its theoretical bases were given.. At the end of the paper is shown an implemented software package that allows the implementation of the proposed hybrid system for cryptographic protection of files. The software is implemented in the environment of Microsoft Visual Studio 6.0 IDE.

Keywords: cryptography, protection of files, symmetric and asymmetric code systems, computer implementation

SADRŽAJ

Sažetak	3
<i>Abstract</i>	3
SADRŽAJ	4
1 UVOD.....	- 1 -
1.1 Šifrovanje.....	- 4 -
1.2 Ključevi i algoritmi	- 5 -
2 RSA metoda.....	- 9 -
2.1 RSA-Istorijski osvrt.....	- 9 -
2.2 Postupak RSA metoda na primeru dva korisnika	- 9 -
2.3 RSA algoritam.....	- 10 -
2.4 Elementarni principi teorije velikih brojeva	- 10 -
2.4.1 Prosti brojevi i faktORIZACIJA	- 10 -
2.4.2 Kongruencija	- 11 -
2.5 "Uprošćen" primer izračunavanja ključeva i šifrovanja-dešifrovanja poruke	- 12 -
3 RC4 metoda	- 15 -
3.1 RC4-Istorijski osvrt	- 15 -
3.2 Postupak RC4 metoda	- 15 -
4 Hibridni sistem.....	- 18 -
4.1 Hibridni sistem realizovan u Microsoft Visual Studio 6.0 IDE	- 18 -
4.2 Opis procesa koji se obavljaju u hibridnom sistemu	- 18 -
4.2.1 Proces komunikacije prilikom slanja zahteva za dostavu RSA javnog ključa... -	20 -
4.2.2 Proces komunikacije prilikom kreiranja RSA javnog ključa i privatnog ključa i njegovo slanje na e-mail	- 21 -
4.2.3 Proces komunikacije nakon prijema RSA javnog ključa, šifrovanja fajla i šifrovanja RC4 simetričnog ključa.....	- 22 -
4.2.4 Proces komunikacije prilikom prijema šifrovanog fajla i dešifrovanje istog	- 23 -
4.3 Neka rešenja problema koji su se javljali prilikom izrade ovog hibridnog sistema. -	24 -
4.3.1 Programiranje matematike velikih brojeva	- 24 -
4.3.2 Bitsko programiranje	- 24 -
4.3.3 Heksadecimalno programiranje.....	- 24 -
4.3.4 Programiranje "mešanja" za RC4 algoritam	- 25 -
5 Zaključak	- 26 -
6 Literatura i Internet linkovi.....	- 27 -

1 UVOD

U ovom radu prikazane su i analizirane tehnike za digitalno šifrovanje fajlova, a zatim je data jedna praktična realizacija. Ona se zasniva na simetričnim i asimetričnim šifarskim sistemima. Posebna pažnja je posvećena RSA i RC4 algoritmima, čijom se zajedničkom primenom dobija jedan hibridni sistem šifrovanja i dešifrovanja podataka.

Šifrovanje i dešifrovanje fajlova je klasična aplikacija za kojom od davnina postoji značajna potreba. Ideja je da se izvrši šifrovanje fajlova, a zatim da se iskoristi bilo koja klasična tehnika (servis) za distribuciju tih fajlova. Sa stanovišta realizacije ovo je najjednostavnija tehnika jer se ne razmatra i ne realizuje distribucija fajlova kao poseban problem u prenosu podataka.

Standardni programi za zaštitu fajlova na aplikativnom nivou obično imaju sledeće funkcionalnosti: izbor algoritma za šifrovanje, izbor radnog ključa, unošenje i brisanje ključeva, šifrovanje za jednog korisnika ili za grupu korisnika i sl. Standardno ovakvi programi šifruju sve vrste fajlova, nezavisno iz kog su programa nastali. To se postiže tako što se fajlovi posmatraju kao niz bajtova, a uopšte nije bitan sadržaj. Šifrovanjem se dobija zaštićeni fajl sa posebnom ekstenzijom. Zbog jednostavnosti korišćenja značajno je da pri dešifrovanju postoji automatizam. Ako bi se i ovde birali parametri šifrovanja veoma lako bi mogla da nastupi konfuzija tj. pogrešno dešifrovanje.

Rad je podeljen u pet poglavlja.

U prvom poglavlju se razmatraju teorijske osnove šifrovanja podataka. Posebna pažnja je posvećena algoritmima i šifarskim ključevima. U klasičnoj kriptografiji, koja počiva na simetričnim šifarskim sistemima, sva tajnost šifarskog sistema se zasniva na tajnosti ključeva. Algoritmi moraju da ispoštuju niz matematičkih zahteva, ali njihova tajnost nije potrebna. Proizilazi da su tajniji šifarski sistemi oni u kojima se koristi duži ključ. Naravno, dužina tajnog simetričnog šifarskog ključa je uvek ograničena i uslovljena je mogućnostima računarske simulacije i na kraju praktične realizacije.

Poglavlje dva se odnosi na asimetrične šifarske sisteme. Posebno je analiziran, a kasnije primenjen RSA šifarski algoritam. Suština ovog algoritma je danas dobro poznata. Tajnost ovakvih sistema se ne zasniva na tajnosti ključeva već na složenosti izračunavanja privatnih ključeva na osnovu javnih ključeva. I ovde je jasno pravilo da su tajniji sistemi koji imaju duže ključeve. Poseban problem koji je ovde razmatran je računarska implementacija ovakvih algoritama i rad sa velikim prostim brojevima. Dati su praktični primeri i ukazano je na složenost ovog problema.

U trećem poglavlju se konkretno razmatra teorijska osnova i sam praktičan rad RC4 algoritma, koji je jedan od najrasprostranjenijih simetričnih šifarskih algoritama. Ono što se prvo uočava je da je rad ovog algoritma izuzetno brz. To se postiže šifrovanjem osnovnog teksta bajt po bajt na principu permutacije bajtova. Za permutaciju se koristi ključ varijabilne dužine (1 -256 bita). Ova njegova osobina, dakle brz rad zasnovan na relativno jednostavnom rešenju, je i osnovni razlog zbog čega je ovaj algoritam najčešće u praktičnoj upotrebi.

Četvrto poglavlje objašnjava samu suštinu ovog rada, odnosno praktičnu realizaciju jednog hibridnog sistema za šifrovanje, koji objedinjuje asimetrične i simetrične algoritme za šifrovanje. Svaki hibridni sistem mora pre svega da obezbedi maksimalnu tajnost ali i brzinu u radu, zbog čega i kombinuje asimetrični sa simetričnim sistemom.

Praktična realizacija ovog hibridnog sistema ostvarena je u okruženju Microsoft Visual Studio 6.0 IDE, pri čemu je posebna pažnja posvećena jednostavnosti korišćenja, pri čemu se naročito misli na proces dešifrovanja šifrovanih fajlova. Jednostavnost je postignuta i

izborom samog IDE okruženja za izradu aplikacija. Neke funkcionalnosti, poput izbora algoritma za šifrovanje ili distribucija šifrovanih fajlova, u ovom radu su izbegnute, opet zarad jednostavnosti rada.

Ono što bi moglo da se navede kao određeni doprinos ovog rada, bi bila jednostavnost rada programa koji je praktična realizacija hibridnog sistema, pri čemu je posebna pažnja usmerena na korišćenje dužih ključeva RSA algoritma (dužinu ključa bira korisnik aplikacije), što u kombinaciji sa brzim radom RC4 algoritma, daje prilično dobru funkcionalnost i efektivnost rada programa za hibridno šifrovanje i dešifrovanje fajlova.

I DEO – ŠIFROVANJE, ALGORITMI I KLJUČEVI

1.1 Šifrovanje

Kada je pismo postalo sredstvo komunikacije, pojavila se potreba da se neka pisma sačuvaju od tuđih pogleda. Tada je i kriptografija ugledala svetlost dana. Od samog početka, šifrovanje podataka koristilo se prvenstveno u vojne svrhe. Jedan od prvih velikih vojskovođa koji je koristio šifrovane poruke bio je Julije Cezar. Naime, kada je Cezar slao poruke svojim vojskovođama, on je te poruke šifrovao tako što su sav ili pojedina slova u tekstu bila pomerana za tri, četiri ili više mesta u abecedi. Takvu poruku mogli su da dešifruju samo oni koji su poznavali „pomeri za“ pravilo. Poznata Cezarova izjava prilikom prelaska Rubikona u šifriranom dopisivanju glasila bi: *fqkf ofhzf kyz*. Pomicanjem svakog slova za šest mesta u abecedi lako se može pročitati pravi smisao poruke: *Alea iacta est* (kocka je bačena) .

Prvu poznatu raspravu o kriptografiji napisao je na 25 stranica italijanski arhitekta Leone Batista Alberti 1467. godine. On je takođe tvorac takozvanog šifarskog kruga i nekih drugih rešenja dvostrukog prikrivanja teksta koja su u XIX veku prihvatili i usavršavali nemački, engleski i francuski šifrantski biro.

Pola veka nakon toga objavljeno je u pet svezaka delo Johanesa Trithemusa prva knjiga iz područja kriptografije. U 16. veku značajan doprinos daju milanski doktor Girolamo Kardano, matematičar Batisto Porta i francuski diplomata Blaise de Vigenere.

Sve do Drugog svetskog rata šifrovane poruke mogle su se koliko-toliko i dešifrovati. Na nemačkoj strani pojavila se mašina koja je šifrovala poruke na do tada još neviđen način. Nemci su mašinu nazvali Enigma (sl. 1.1.). Međutim, ma koliko je ona u to vreme bila revolucionarna, saveznici su uspeali da razbiju poruke šifrovane Enigmom.



Slika 1-1.1. Enigma u saobraćajnom muzeju u Lucernu, Švajcarska

Ogroman doprinos Američkoj pobedi nad Japancima na Pacifiku dali su Navaho šifranti. Jezik Navaho indijanaca znalo je svega nekoliko ljudi van Navaho plemena a medju njima nije bilo nijednog Japanca. Japanci za koje nije bilo tajni u engleskom jeziku ostali su nemoćni pred ovakvim načinom šifriranja poruka.

Posle Drugog svetskog rata i pojavom prvih računara otvorila su se nova vrata kriptografiji. Računari su vremenom postajali sve brži i brži, radeći i po nekoliko stotina, a kasnije i miliona operacija u sekundi. Novom brzinom rada je omogućeno probijanje šifri za sve manje vremena. Uporedo s tim, radilo se i na izmišljanju novih, sigurnijih i komplikovanijih algoritama za šifrovanje.

Sama kriptografija, odnosno veština smišljanja šifara, sa kriptanalizom, koja je veština razbijanja šifara, čini deo posebne discipline informatičke nauke, koja se zove kriptologija.

Pojmovi šifrovanje odnosno kodiranje su za profesionalce kriptologe različiti, mada su se istorijski gledano, dosta koristili za iste radnje, odnosno stvari, dok se danas konkretno razlikuju. Šifra omogućava zamenu znak za znak (bit za bit), dok se kôdom jedna reč zamenjuje drugom rečju ili simbolom.

Termin kôd se u informatici danas pre svega, koristi za navođenje posebnih sintaksnih i semantičkih pravila u vidu službenih reči, koje programeri koriste prilikom programiranja raznih aplikacija u određenom IDE okruženju.

Kada je termin šifra u pitanju, misli se pre svega na transformaciju poruke poznate kao osnovni tekst, koja se pomoću nekih funkcija, pre svega matematičkih, čiji su parametri zadani ključem, transformiše u šifrovani tekst. Ovakav šifrovani tekst se može nazivati šifrom.

Šifre sa olovkom i papirom koje su korišćene u prošlosti su poznate pod nazivom "klasična šifra". One uključuju proste "**šifre zamene** (supstitucije)" i "**šifre pomeranja** (transpozicije)". Na primer, "dobar dan" može da se šifrue kao "ŽSDGĆ ŽGP", gde se u poruci "d" zamenjuje sa "Ž", "a" sa "G" itd. Pomeranjem bi "dobar dan" moglo da bude "RDANDOBA". Ove proste šifre i primere je vrlo lako razbiti, čak i bez pomoći parova otvoreni tekst - šifrovani tekst.

Proste šifre su zamenjene šiframa "**polialfabetske zamene**", koje menjaju alfabet zamene za svako slovo. Na primer, "dobar dan" može da se zameni sa "ŽĆZIA ONV", gde se u poruci "d" zamenjuje sa "Ž" i "O", "a" sa "I" i "N". Čak i na malom uzorku poznatog ili pretpostavljenog otvorenog teksta, šifre polialfabetske zamene i šifre pomeranja slova (namenjene za šifrovanje olovkom i papirom) se vrlo lako razbijaju.

Početak dvadesetog veka izumljene su elektro-mehaničke mašine za šifrovanje i dešifrovanje, a koristile su pomeranje, polialfabetsku zamenu i neku vrstu "dodatne" zamene. U "rotor mašinama", nekoliko rotirajućih diskova je služilo za polialfabetsku zamenu, a dodatne ploče su služile za neku drugu zamenu. Ključevi su se lako menjali zamenom rotirajućih diskova i prespajanjem žica dodatnih ploča. Iako su ovi metodi šifrovanja bili kompleksniji od prethodnih i zahtevali mašine za šifrovanje i dešifrovanje, izumljene su druge mašine (kao npr. britanska "Bomba") za dešifrovanje ovih metoda šifrovanja.

1.2 Ključevi i algoritmi

Pojam ključa je dosta važan, jer se ključem, koji je u stvari srazmerno kratak tekstualni niz, bira jedan od više mogućih načina šifrovanja.

Kriptografski algoritmi zasnovani su na matematičkoj funkciji koja se koristi za šifrovanje i dešifrovanje. Razlikuju se dve vrste algoritama:

- ograničeni algoritmi: bezbednost se zasniva na tajnosti algoritma (istorijski interesantni).
- algoritmi zasnovani na ključu: bezbednost se zasniva na ključevima, a ne na detaljima algoritma koji se može publikovati i analizirati (algoritam je javno poznat, a ključ se čuva tajnim).

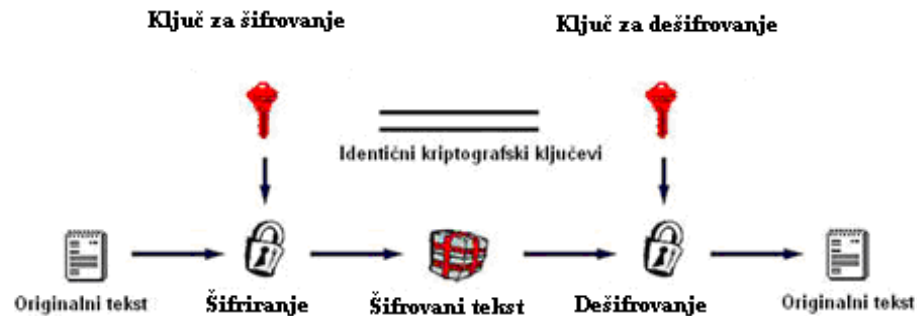
Kerkofov princip da svi algoritmi moraju biti javni, dok su samo ključevi tajni, pojašnjava važnost ključa, jer se u kriptografiji polazi od pretpostavke da kriptanalitičar zna metode korišćenja šifrovanja i dešifrovanja koje se nazivaju algoritmima šifrovanja i dešifrovanja ali ne može da zna tajne parametre definisane ključem. Dakle, bez obzira na

složenost algoritama, kriptanalitičar ih može poznavati, ali tajnost leži isključivo u ključevima.

Moderni metodi šifrovanja mogu da se podele po dva kriterijuma: po tipu korišćenog ključa i tipu ulaznih podataka.

Po tipu ključa, šifre se dele na:

- Algoritmi simetričnog ključa (kriptografija sa tajnim ključem- sl. 1-2.1.), gde se isti ključ koristi i za šifrovanje i za dešifrovanje, i



Slika 1-2.1. Simetrični kriptografski sistem sa tajnim ključem

- Algoritmi asimetričnog ključa (kriptografija sa javnim ključem- sl. 1-2.2.), gde se koriste dva različita ključa za šifrovanje i dešifrovanje.



Slika 1-2.2. Asimetrični kriptografski sistem sa javnim ključem

Kod algoritma simetričnog ključa (npr. *DES* и *AES*), pošiljalac i primalac moraju da imaju isti ključ, koji je unapred pripremljen i koji mora da bude tajan za sve kojima nije namenjen - pošiljalac koristi taj ključ za šifrovanje, a primalac isti taj ključ za dešifrovanje.

Kod algoritma asimetričnog ključa (npr. *RSA*), postoje dva posebna ključa: "javni ključ" se objavljuje i omogućava pošiljaocu da izvrši šifrovanje, dok "tajni ključ" primalac čuva kao tajnu i koristi ga da izvrši ispravno dešifrovanje.

Po tipu ulaznih podataka, šifre se mogu svrstati u dve grupe:

- Šifra blokova, kojom se šifruje blok podataka fiksne dužine, i
- Šifra toka, kojom se šifruje kontinualni niz podataka.

U čisto matematičkom napadu (tj. u nedostatku drugih informacija koje bi pomogle u razbijanju šifre), pre svega su bitna tri faktora:

- Matematičke prednosti da se otkriju i iskoriste slabosti šifre.

- Dostupna snaga izračunavanja, tj. snaga računara koja se može upotrebiti za rešavanje problema. Treba napomenuti da prosečne performanse/kapacitet jednog računara nisu jedini faktor koji se uzima u obzir. Protivnik može, recimo, da koristi više spregnutih računara i da tako značajno ubrza potragu za ključem (takozvani "napad grubom silom").
- Veličina ključa, tj. dužina ključa koji je upotrebljen za šifrovanje poruke. Kako raste veličina ključa, tako se povećava i kompleksnost pretraživanja grubom silom, sve do tačke kad prestaje da bude praktično direktno razbijanje šifre.

Pošto je željen efekt teškoća izračunavanja, teorijski se biraju algoritam i nivo težine, a na osnovu toga se bira odgovarajuća dužina ključa.

Klod Šenon je dokazao, koristeći razmatranja teorije informacija, da svaka teoretski neprobojna šifra mora imati dužinu ključa najmanje koliko je dugačak otvoreni tekst i da se ključ koristi samo jednom (eng. *one-time pad*).

II DEO – RSA METODA

2 RSA metoda

Savremena kriptografija tradicionalno vrši podelu kako metoda šifrovanja, tako i metoda korišćenja ključeva, tako da RSA metoda spada u kategoriju supstitucione šifre algoritma za šifrovanje javnim ključem. Supstituciona, znači da se svaki znak ili grupa znakova, šifruje tako što se zamenjuje drugim znakom ili grupom znakova, a termin javni ključ, znači da se i algoritam za šifrovanje i ključ objave javno, dok se dešifrovanje obavlja privatnim ključem koji zadržava tajnost. Dakle, koriste se dva ključa, javni koji koristi svako ko želi da šifruje i privatni koji služi za dešifrovanje. Ovo je jedna od najkomplikovanijih metoda za kriptanalizu ali ima dosta spor rad.

2.1 RSA-Istorijski osvrt

Ono što je kriptologe dugo mučilo i što je predstavljalo najslabiju tačku kriptosistema, je bilo distribuiranje ključeva. Bolje rečeno, nije se u početku znalo kako zaštititi ključ od krađe. Bilo je obavezno da korisnici kriptosistema koriste isti ključ za šifrovanje i dešifrovanje, ili makar da koriste dva ključa koja se lako izvedu jedan iz drugoga. Dakle, morao se ključ ili ključevi podeliti svim korisnicima sistema. Što više korisnika kriptosistema, veća mogućnost da se neovlašćeno dođe do ključa koji se ne može držati u sefu. Problem je bio u unutrašnjoj slabosti kriptosistema, jer je lopov mogao lako doći do ključa zbog velikog broja korisnika, od kojih makar jedan može biti "slaba" karika sistema.

Dva istraživača sa Stanford Univerziteta, Diffie i Hellman su 1976. godine predložili novu vrstu kriptosistema kod kog se koriste različiti ključevi za šifrovanje i dešifrovanje, koji se **ne mogu** lako izvesti jedan iz drugoga.

Ceo sistem se zasniva na ispunjenju tri zahteva:

1. $D(E(P))=P$,
2. D se izuzetno teško može izvesti iz E,
3. E se ne može provaliti napadom zasnovanim na šifrovanju osnovnog teksta.

Algoritmom E se šifruje osnovni tekst P a algoritmom D se dešifuje dobijeni šifrovani tekst E(P).

Prvi zahtev govori da treba biti omogućeno lako dešifrovanje šifrovanog teksta i brzo i tačno dobijanje osnovnog teksta.

Drugi zahtev je u stvari bilo najteže rešiti i on predstavlja srž problema sa kojima su se susretali prethodni kriptolozi.

Treći zahtev dozvoljava da kriptanalitičar eksperimentiše sa algoritmom koliko hoće ali da ne može da dođe do osnovnog teksta.

Ispunjenjem ovih uslova dobijamo mogućnost da ključ za šifrovanje može biti javan.

2.2 Postupak RSA metoda na primeru dva korisnika

U kriptografiji su najpopularnija imena za korisnike kriptosistema Alisa i Bob.

Recimo da Alisa želi da razmenjuje tajne poruke sa Bobom. Prvo će morati da generiše dva algoritma koja će zadovoljavati zahteve koje su definisali Diffie i Hellman, što znači, da će moći da javno objavi svoj algoritma i ključ za šifrovanje, na primer na svojoj Web strani. Zbog same činjenice da je ključ dostupan svim posetiocima sajta, ovaj

se postupak naziva šifrovanje javnim ključem, eng. public-key cryptography. Sam ključ i algoritam podešen parametrima ključa, mogu se zajednički obeležiti sa E_a .

Sa D_a će se obeležiti ključ i algoritam podešen Alisinim privatnim ključem, koji će služiti za dešifrovanje i koji Alisa čuva u tajnosti.

Bob mora sve to isto da uradi i da nakon toga objavi svoj javni ključ i kod E_b a u tajnosti da čuva svoj privatni ključ i kod za dešifrovanje D_b .

Dakle i E_a i D_b ključ za šifrovanje moraju se nalaziti u javno dostupnim datotekama.

Kada Alisa napiše svoju poruku(P), izračunava $E_b(P)$ i dobijeni rezultat šalje Bobu. Bob koristi svoj privatni (tajni) ključ D_b kako bi pristupio izračunavanju $D_b(E_b(P))=P$ i naravno, uspešno dolazi do osnovnog teksta P. Skoro sigurno je nemoguće izračunati D_b znajući E_b , a bez D_b se ne može ni pokrenuti relacija $D_b(E_b(P))=P$, dakle i da se dođe do $E_b(P)$, odnosno šifrovanog osnovnog teksta, dešifrovanje je nemoguće. Bob, sada siguran, piše odgovor R, računa $E_a(R)$ i tako šifrovan odgovor šalje Alisi. Dakle, komunikacija je bezbedna.

2.3 RSA algoritam

Algoritam koji je ispunjavao sva tri gore navedena zahteva i koji je omogućavao gore opisani postupak komuniciranja Alise i Boba je RSA algoritam, a čiju je osnovnu tehniku 1973. godine otkrio Clifford Cocks. Ova se tehnika čuvala u tajnosti sve do pojave jedne grupe istraživača sa Masačusets tehničkog instituta 1977. godine. Ron Rivest, Adi Shamir i Len Adleman te godine istražuju jednu dobru metodu koja se zasnivala na osnovnim postulatima koje je postavio Cocks, a 1978. godine, nezavisno od Cocks-ovog rada, je i objavljuju pod imenom RSA (početni inicijali njihovih imena).

Osnovna metoda RSA algoritma se zasniva na određenim principima teorije velikih brojeva i izdržala je sve pokušaje provaljivanja više od 30 godina i smatra se veoma otpornom.

Upravo ova otpornost na napade, stvara i osnovnu manu metode, a to je dosta spor rad sa njom, jer se zahtevaju dugački ključevi (minimum 1024 bita, preko 100 decimalnih cifara).

2.4 Elementarni principi teorije velikih brojeva

2.4.1 Prosti brojevi i faktORIZACIJA

Da bi se RSA algoritam razumeo, potrebno je upoznati neke osnovne pojmove iz teorije brojeva. Pre svega, ovaj algoritam radi sa veoma velikim prostim brojevima.

Po Definiciji1: **Prost broj je broj koji nije deljiv niti sa jednim brojem, osim sa 1 i sa samim sobom. Za broj koji nije prost, kažemo da je složen.**

Teorema1: Prostih brojeva ima beskonačno mnogo.

Dokaz1: Ako pretpostavimo da prostih brojeva ima konačno mnogo, te da su oni: $p_1, p_2, \dots, p_n$. Od njih se može napraviti broj $n = p_1 p_2 \dots p_n + 1$. Očigledno da taj broj nije deljiv ni sa jednim brojem iz skupa $p_1, p_2, \dots, p_n$, jer pri deljenju sa svakim od njih daje ostatak 1. Dakle, ovaj broj n je ili deljiv sa nekim prostim brojem koji ne leži u skupu brojeva $p_1, p_2, \dots, p_n$, ili je sam prost broj. Ovo je kontradikcija sa polaznom pretpostavkom dokaza, da

smo skupom $p_1, p_2, \dots, p_n$ obuhvatili sve proste brojeve, dakle dokazujemo da je teorema tačna, da ima beskonačno mnogo prostih brojeva.

Lema1: Ako neki prost broj p deli umnožak dva cela broja a i b ($p|ab$), onda on deli i same te brojeve posebno ($p|a$; $p|b$).

Teorema2: Svaki prirodan broj $n > 1$ se može prikazati kao umnožak jednog ili više prostih brojeva, tj. u obliku $n = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n}$, što se naziva faktORIZACIJOM broja n . FaktORIZACIJA broja je jedinstvena (do na raspored faktora u umnošku).

Dokaz2: Pretpostavimo namerno da je neki broj n , gde je $n > 1$, najmanji pozitivan broj koji se ne može prikazati kao umnožak prostih brojeva. On ne može biti prost, jer tada je on po Teoremi2 umnožak sa samo jednim članom. Dakle, ako zapišemo da je $n = cd$, pri čemu su $c, d < n$, a pretpostavili smo da je n najmanji prirodan broj koji se ne može faktorizovati, onda se c i d mogu prikazati kao umnožak prostih brojeva. Međutim umnožak prostih brojeva pomnožen sa nekim drugim (ili istim) umnoškom prostih brojeva, daje umnožak prostih brojeva, pa zaključujemo da je n ipak umnožak prostih brojeva.

Sada pretpostavimo da se ovaj broj n može zapisati kao dva različita umnoška prostih brojeva, $n = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n} = q_1^{b_1} q_2^{b_2} \dots q_n^{b_n}$. Sada ćemo pretpostaviti da su svi p i q različiti, jer ako ima istih faktora sa leve i desne strane, možemo jednačinu podeliti s tim faktorima, dok ne dobijemo sve različite faktore. Tada, pošto p_1 deli levu stranu jednačine, mora deliti i desnu, pa prema Lemi1, deli jedan od faktora q_n . Međutim, kako su svi q_n prosti, sledi da je p_1 jednak nekom q_k . Dakle, ako se p_1 sa leve strane ponavlja a_1 puta, onda sa desne strane jednačine mora postojati jednak broj faktora q_k . Slično se može smatrati za sve p -ove sa leve strane, iz čega vidimo da se sa obe strane znaka jednakosti nalaze potpuno isti faktori. Time se obara polazna pretpostavka da se n može zapisati kao dva različita umnoška prostih brojeva i dokazuje Teorema2.

Po Definiciji2: Neka su a i b celi brojevi i n najveći ceo broj koji deli a i b . Tada se n naziva *najveći zajednički delitelj* brojeva a i b i označava se sa $\gcd(a, b)$. Ako je $\gcd(a, b) = 1$, kažemo da su a i b relativno prosti brojevi. Ako sada sa $\Phi(n)$, označimo funkciju koja daje broj prirodnih brojeva manjih od n , a koji su relativno prosti sa n , možemo na primeru pokazati Eulerovu funkciju: $\Phi(10) = 4$. Znajući da je $\gcd(x, 10) = 1$ za svako $x = 1, 3, 7, 9$. Očito je da je $\Phi(n) > 1$, obzirom da svaki celi broj ima barem jednog delitelja (broj 1).

Ako je p prost broj, očigledno da je $\Phi(p) = p - 1$. Ako je pak $n = pq$, umnožak dva prosta broja, tada su sa n relativno prosti brojevi svi brojevi manji od njega a koji u sebi ne sadrže faktore p i q . Takvih je $\Phi(p)\Phi(q) = (p-1)(q-1)$.

2.4.2 Kongruencija

Pretpostavimo da su a i b dva cela broja a n ceo broj različit od nule. Ako je $n|(a-b)$, kažemo da su a i b kongruentni modulo n i to zapisujemo na sledeći način:

$$a \equiv b \pmod{n}$$

Drugi način zapisivanja te relacije je:

$a = b + nk$, pri čemu je k ceo broj. Ako važi $a \equiv b \pmod{n}$ i $c \equiv d \pmod{n}$, korišćenjem $a = b + nk$, lako se proveravaju sledeće tvrdnje:

ako je $a \equiv b \pmod{n}$, onda je $b \equiv a \pmod{n}$, $a + c \equiv b + d \pmod{n}$, $a - c \equiv b - d \pmod{n}$, $ac \equiv bd \pmod{n}$. Kada je deljenje u pitanju, ako je $\gcd(a, n) = 1$, za $ab \equiv ac \pmod{n}$, sledi $b \equiv c \pmod{n}$.

Iz ovoga se može izvesti još jedna važna lema.

Lema2: Ako je $a \equiv bc \pmod{n}$, onda je $a \equiv (b \pmod{n})(c \pmod{n}) \pmod{n}$.

Ova lema generiše Teoremu3 ili Eulerovu teoremu, koja igra najvažniju ulogu prilikom dekriptovanja osnovnog teksta, jer zbog ogromnog eksponenta koji se koristi u paru privatnog ključa, bilo bi skoro nemoguće doći do osnovnog teksta, tako da se uz pomoć ove leme problem računanja ostatka nakon potenciranja velikim, svodi na određivanje ostatka potenciranjem dvostruko manjim brojem. Postupak se sprovodi sve dok se eksponent ne svede na 1. Algoritam je poznat pod nazivom "modular exponentiation".

Sama Lema2 dokazuje Teoremu3, tako što se "širi" zapis

$a \equiv (b(\bmod n))(c(\bmod n))(\bmod n)$, konkretno desne strane i dobija se:
 $a \equiv (b+in)(c+jn)(\bmod n) = (bc+bjn+cin+ijn^2)(\bmod n) = bc+bjn+cin+ijn^2+kn = bc+n(bj+ci+ijn+k)$.
 Vidimo da uz n stoji opet neki ceo broj, pa je poslednji izraz jednak $bc(\bmod n)$.

Iz svega ovoga se može izvesti Teorema3 ili Eulerova teorema:

$1 \equiv a^{\Phi(n)} \bmod(n)$, ako je $\gcd(a,n)=1$.

Na osnovu ovih elementarnih principa matematike, objašnjava se teorija velikih brojeva, konkretno težina faktORIZACIJE velikog broja.

Ipak, dosta je teško na osnovu komplikovanih matematičkih zapisa "plastično" shvatiti ovu teoriju, zato se pristupa objašnjenju teorije koristeći relativno male zapise brojeva, sa relativno malim brojem cifara.

2.5 "Uprošćen" primer izračunavanja ključeva i šifrovanja-dešifrovanja poruke

Za početak je potrebno izabrati dva prosta broja:

$$\begin{aligned} p &= 61 \\ q &= 53 \end{aligned}$$

Njihovim množenjem dobija se broj $n=pq=3233$

Na osnovu Eulerove funkcije, računamo $\Phi(n)=z=(p-1)(q-1)=3120$

Nakon ovog izračunavanja, bira se eksponent šifrovanja, koji je prvi broj u paru brojeva za javni ključ (e,n) : na primer $e=17$.

Sada se može pristupiti izračunavanju eksponenta dešifrovanja, koji je prvi broj u paru brojeva za privatni ključ (d,n) , na osnovu formule $de \equiv 1(\bmod \Phi(n))$
 $d=2753$

Sada postoje i javni $(17,3233)$ i privatni ključ $(2753,3233)$, pa se može objasniti princip šifrovanja i dešifrovanja poruke RSA metodom, na sledeći način:

Želimo na primer šifrovati poruku, odnosno broj 123. Da bi se to uradilo, računamo: $c \equiv m^e \bmod(n) \equiv 123^{17} \bmod(3233) = 855$

Dešifrovanje se vrši na sledeći način:

$m \equiv cd \bmod(n) \equiv 855^{2753} \bmod(3233)$, tako da se ovim izračunavanjem dobija broj 123, što je bila i polazna poruka.

Jasno je, da bi korišćenjem malih prostih brojeva $(p \text{ i } q)$, koji pomnoženi daju broj n i dalje, generišu parove javnog i privatnog ključa, ovo bilo vrlo lako rešivo za

kriptoanalitičare, jer bi mogli brzo da urade faktORIZACIJU broja n , koji je vrlo mali sa svega 4 cifre, pa bi razlaganjem ovog broja lako našli brojeve p i q , a nakon toga bi lako došli i do broja z , odnosno $\Phi(n)$, tako da imajući broj z i eksponent šifrovanja koji je javan, lako pomoću Euklidovog algoritma mogu doći i do tajnog eksponenta dešifrovanja.

III DEO – RC4 METODA

3 RC4 metoda

RC4 metoda spada u kategoriju supstitucione šifre algoritma za šifrovanje simetričnim ključem. Simetričnim znači da se i za šifrovanje i za dešifrovanje koristi isti ključ, koji mora biti dostupan onome ko šifrue podatke i onome ko ih dešifrue, a opet, mora zadžati ono glavno svojstvo kriptografije ključeva, koje je dato u vidu Kerkofovog principa, a to je tajnost. Ovo je metoda koja ima izuzetno brz rad ali je veoma laka za kriptanalizu.

3.1 RC4-Istorijski osvrt

Profesor Ronald Lorin Rivest (rođen 1947, Šenestedi, Njujork) je kriptograf, i profesor kompjuterskih nauka na MIT-u, na katedri za elektrotehniku i računarstvo (CSAIL). Najpoznatiji je po svom radu na javnom-ključu enkripciji sa Len Edlmenom i Adijem Šamirom, posebno na RSA algoritmu, za koji su 2002. dobili Tjuringovu nagradu. Član je odbora za tehničko vođenje i razvoj pri Izornoj pomoćnoj komisiji (EAC), čija je svrha pomoć EAC u izradi plana za Dobrovoljni glasački sistem vodiča.

Takođe je pronalazač simetričnog ključa enkripcionog algoritama RC2, RC4, RC5, i ko-pronalazač RC6. „RC“ je „Rivestova brojka“, ili alternativno „Ronov kod“. (RC3 je bio provaljen u RSA obezbeđenju tokom istraživanja, slično RC1 nije nikada objavljen). Takođe je napisao MD2, MD4 i MD5 kriptografske funkcije za pretresanje.

RC4 je algoritam za šifrovanje toka podataka koji koristi varijabilnu dužinu ključa (1 – 256 bita). Ima sledeće karakteristike:

- promenljiva veličina ključa, bajtovski orijentisan algoritam šifriranja toka podataka
- široko rasprostranjen (web SSL/TLS, wireless WEP)
- ključ formira slučajnu permutaciju svih 8-bitnih vrednosti
- koristi se ta permutacija da se šifrira ulazni podatak bajt po bajt

3.2 Postupak RC4 metoda

RC4 se sastoji od 2 zasebna algoritma: KSA (Key Scheduling Algorithm) i PRGA (Pseudo Random Generation Algorithm).

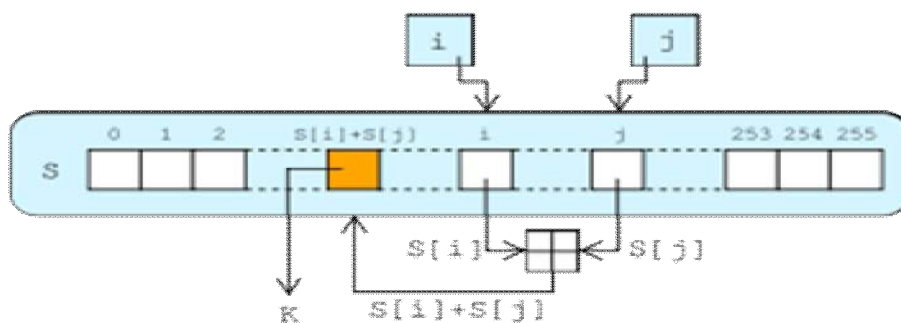
Rad algoritma je jednostavan, dok se istovremeno postižu vrlo velike brzine šifrovanja. Algoritam koristi S kutiju od 256 polja, gdje je svako polje veliko 8 bita. Sadržaj S kutije su permutacije od 0 do 255, dok su permutacije funkcije varijabilnog ključa. Pripremanje S kutije se može prikazati sledećim koracima:

1. Svaki element polja S se inicijalizuje svojim indeksom:
 $S[0] = 0, S[1] = 1, \dots, S[254] = 254, S[255] = 255$
2. Stvori se pomoćno polje S2 iste veličine te se popuni vrednostima ključa K:
 $S2[0] = K[0 \% \text{dužina_K}], S2[1] = K[1 \% \text{dužina_K}], \dots, S2[255] = K[255 \% \text{dužina_K}]$
3. S kutija poprima svoj konačni izgled na sledeći način:
 za $i = 0, i < 256, i++$
 $j = (j + S[i] + S2[i]) \% 255;$
 $\text{temp} = S[i];$
 $S[i] = S[j];$
 $S[j] = \text{temp};$
 S kutija će poslužiti za generisanje pseudoslučajnih brojeva, koji su neophodni za šifrovanje podataka:
 $i = j = 0;$

```
i = ( i + 1 ) % 256 ;
j = ( j + S[i] ) % 256 ;
temp = S[i] ;
S[i] = S[j] ;
S[j] = temp
t = ( S[i] + S[j] ) % 256 ;
K = S[t] ;
```

Za svaki ulazni znak koji se želi šifrovati potrebno je generisati novi pseudoslučajni broj. Šifrovani tekst se dobija operacijom XOR između uvek novo generisanog broja K i svakog ulaznog znaka. Dešifrovanje se obavlja inverznom procedurom, odnosno izvođenjem XOR operacije između generisanog broja K i šifrovanog teksta.

Organizacija za koju je algoritam osmišljen tvrdi da je RC4 otporan na diferencijalnu i linearnu kriptanalizu, ne poseduje kratke cikluse, te je u potpunosti nelinearan (RC4 se može nalaziti u jednom od 21700 mogućih stanja). S kutija se polako razvija sa upotrebom: varijabla i osigurava da se svaki element promeni, dok varijabla j omogućuje da se elementi izmjenjuju pseudoslučajno.



Slika 3-1.1. Šema RC4 algoritma

IV DEO – HIBRIDNI SISTEM ŠIFROVANJA

4 Hibridni sistem

Treba imati u vidu da eksplicitna upotreba simetrične ili asimetrične kriptografije pati od izvesnih nedostataka: Simetrična kriptografija ne raspolaže načinima da se obavi sigurna razmena ključa i to je njena najveća mana. Na drugoj strani, asimetrična kriptografija upotrebom para ključeva kompletno eliminiše opasnost od zloupotrebe presretnutog ključa ali je istovremeno spora (i do nekoliko stotina puta sporija) u poređenju sa konvencionalnim načinom šifrovanja podataka.

Hibridni kriptosistemi koriste i jedan i drugi princip. Osnovni tekst se prvo šifrjuje upotrebom simetričnog ključa (pre toga u nekim slučajevima je bolje izvršiti i dodatnu kompresiju, ako se radi o velikom dokumentu), a zatim se generisani ključ zajedno sa šifrovanim dokumentom pakuje i ponovo šifrjuje primaočevim javnim ključem.

Proces dešifrovanja celog dokumenta je obrnut - primaoc prvo dešifrjuje sa svojim tajnim ključem, pronalazi spakovani simetričan (ponekad nazvan i session ključ, jer je generisan upotrebom Pseudo Random Number Generatora u kombinaciji sa raznim korisnikovim unosima u toku procesa generacije i njegova upotreba se ograničava samo na tu razmenu) ključ i koristi ga da bi dokument vratio u svoj izvorni oblik. Primer uspešnog hibridnog kriptosistema je Phill Zimmermanov Pretty Good Privacy (PGP) paket.

4.1 Hibridni sistem realizovan u Microsoft Visual Studio 6.0 IDE

Hibridni sistem koji je realizovan u ovom IDE okruženju, u osnovi poštuje principe i procese šifrovanja i dešifrovanja prethodno opisane, ali i ima neka malo drugačija rešenja, koja bi trebalo da pruže pre svega komforniji i brži rad aplikacije.

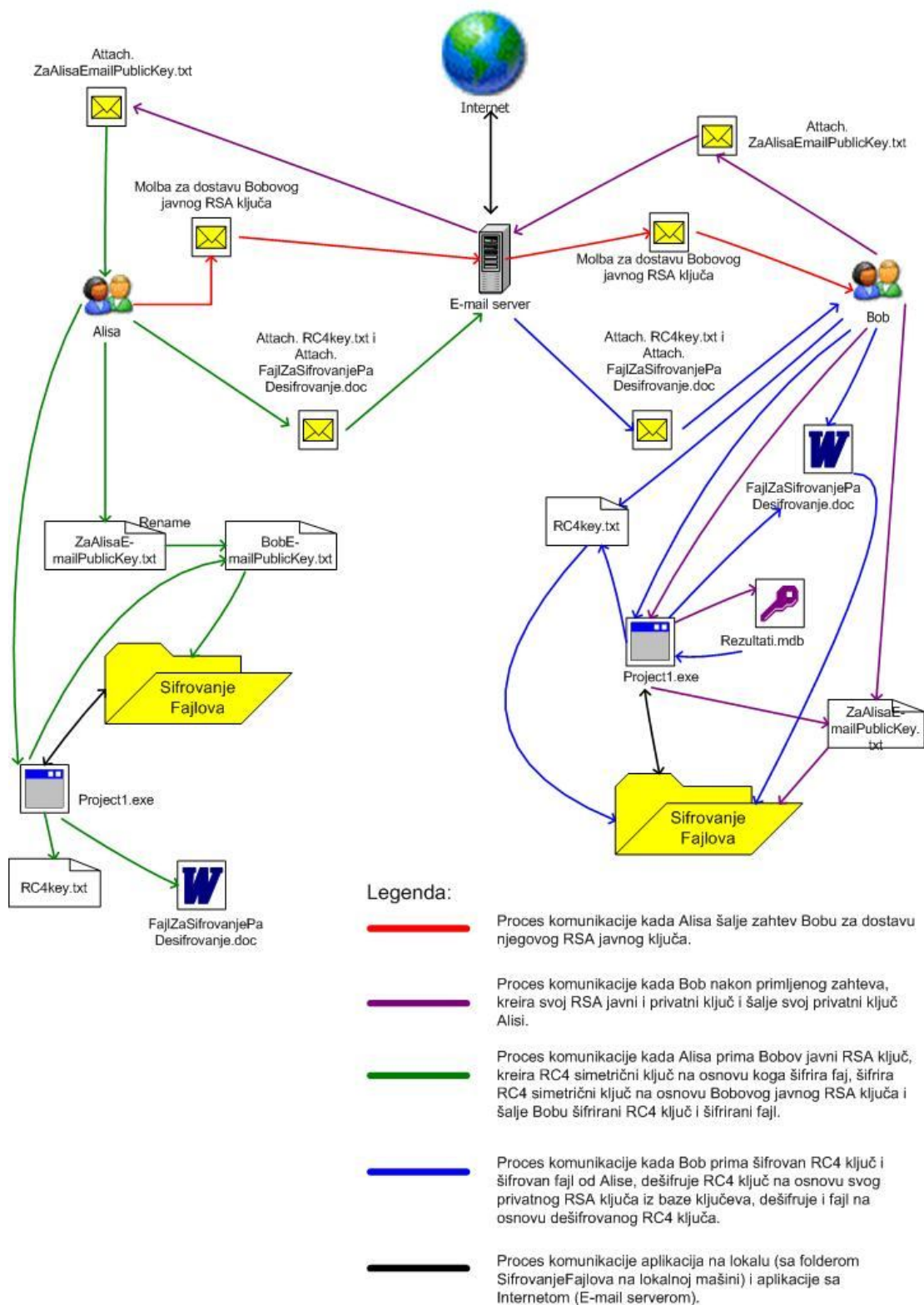
Dakle, aplikacija koristi ključ, nazvan simetrični, koji služi za šifrovanje osnovnog teksta (sadržaja fajla) RC4 algoritmom, a onda se taj ključ posebno šifrira RSA algoritmom, za koji je potreban par varijabli (e, n) koje čine javni ključ, nakon čega se takav šifrovani simetrični ključ i šifrovani fajl predaju onome ko treba da izvrši dešifrovanje. Dakle, izbegnuto je kompresovanje podataka, pri čemu je testirano šifriranje fajla od 7,7MB raznolike sadržine (sa slikama, raznim fontovima i hiper linkovima), sa simetričnim ključem dužine 70 cifara (560 bit-a, može i sa više cifara) i n varijablom iz para javnog ključa dužine 250 cifara (2000 bit-a), pri čemu je brzina rada bila sasvim zadovoljavajuća.

Dešifrovanje ide obrnutim procesom, dakle, prvo se dešifrjuje šifrovani simetrični ključ, a sa njim je moguće dešifrovati šifrovani fajl.

4.2 Opis procesa koji se obavljaju u hibridnom sistemu

Aplikacija urađena u VB6 IDE, koja je data kao prilog ovog rada, obavlja više složenih procesa prilikom šifrovanja i dešifrovanja fajlova. Ovi procesi šematski su prikazani na sl. 4-1.1 i biće detaljno objašnjeni u tekstu koji sledi.

Procesi koje obavlja hibridni sistem za šifrovanje-dešifrovanje

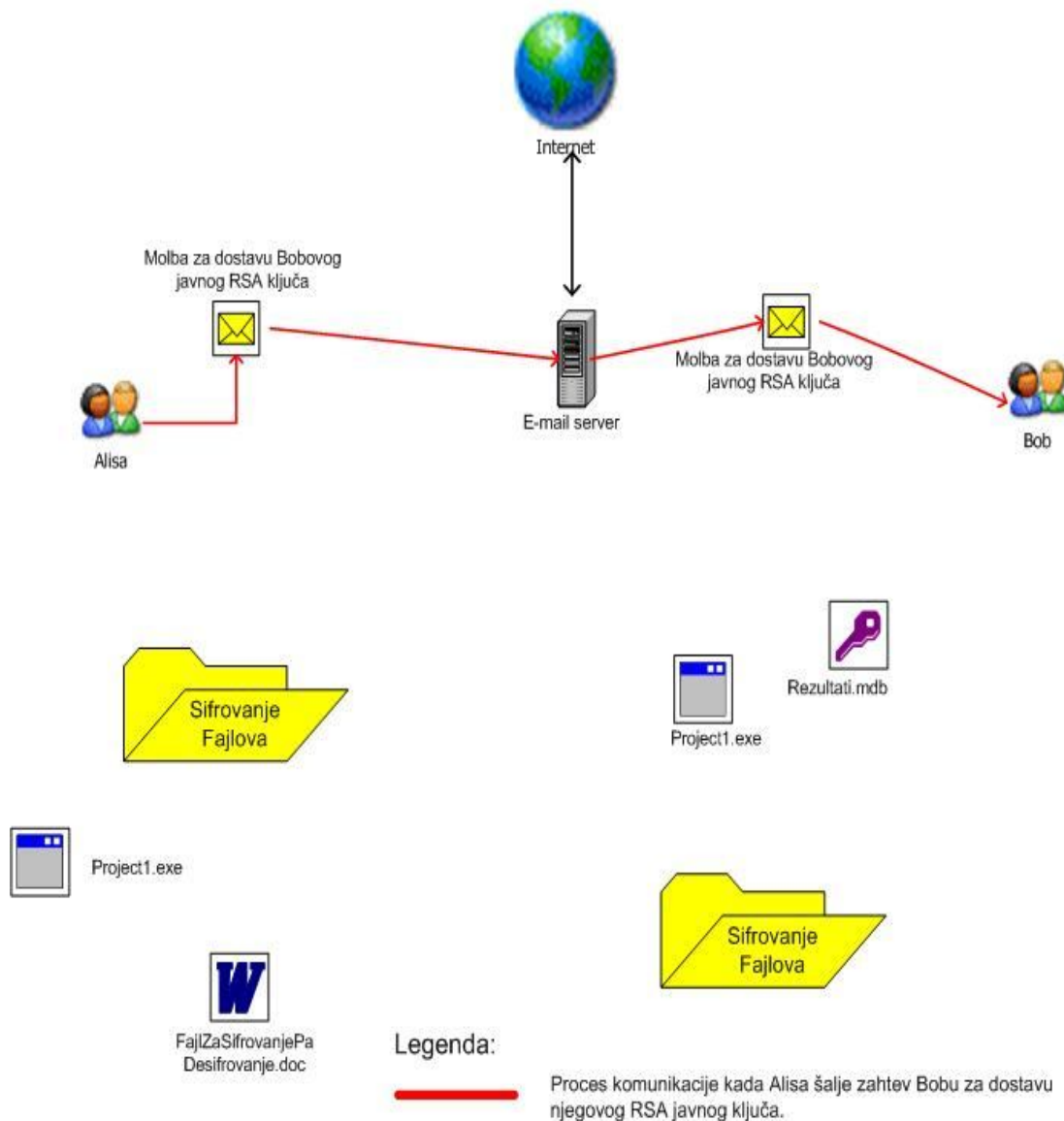


Slika 4-1.1. Procesi koji se obavljaju u hibridnom sistemu

4.2.1 Proces komunikacije prilikom slanja zahteva za dostavu RSA javnog ključa

Kao i u mnogim drugom radovima sa ovom tematikom, koristiće se skoro standardizovana imena Alisa i Bob, za osobe koje žele tajno da razmenjuju podatke, odnosno fajlove.

Proces koji obavlja hibridni sistem za šifrovanje-dešifrovanje prilikom slanja zahteva za dostavu RSA javnog ključa



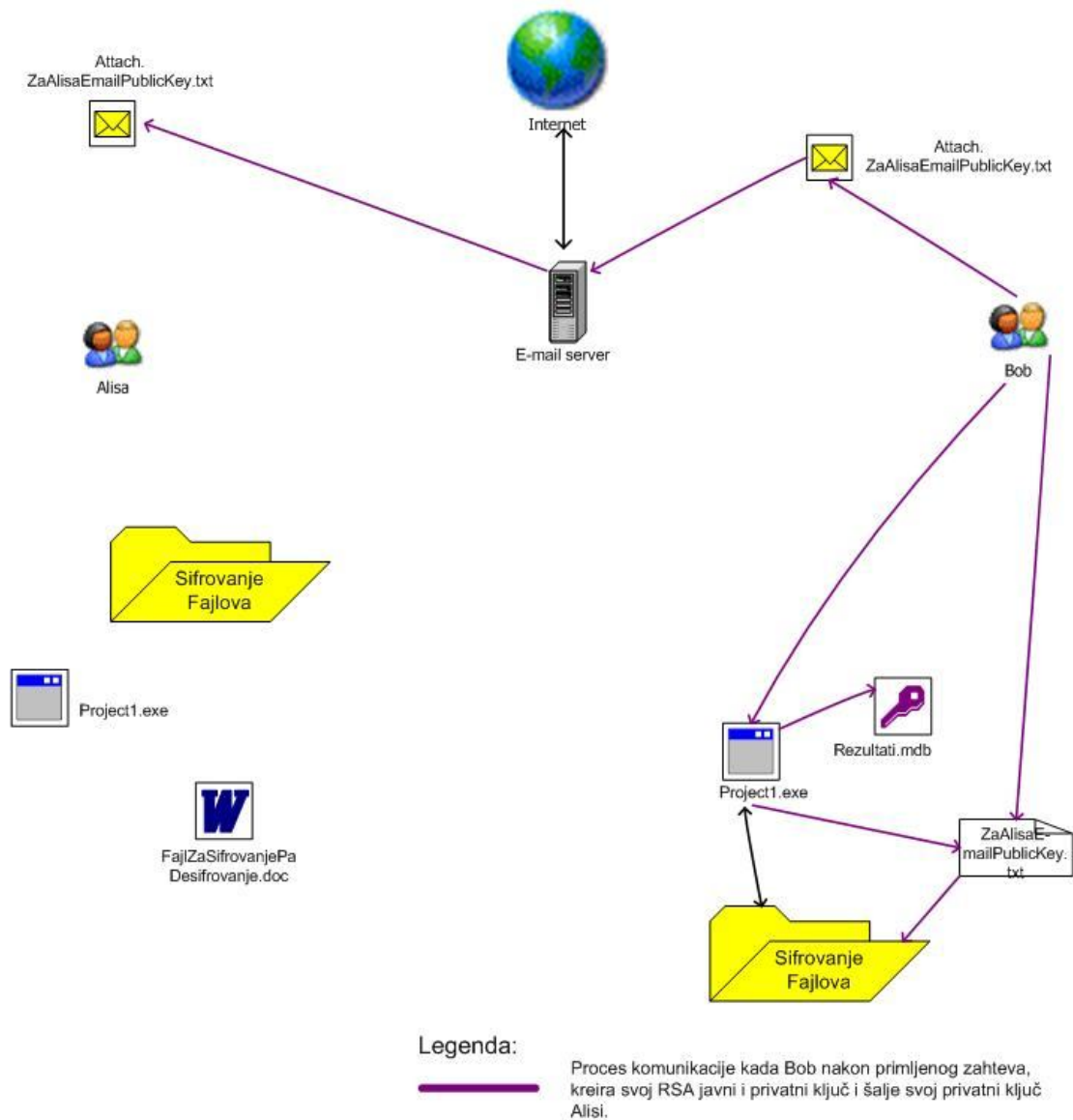
Slika 4-2.1. Prvi proces koji se obavlja u hibridnom sistemu

Sa slike 4-2.1. se može uočiti da onaj ko želi da tajno dostavi (šifrovani) fajl, u ovom slučaju Alisa, drugoj strani, odnosno Bobu, šalje e-mail sa kraćom porukom-molbom da joj se dostavi javni RSA ključ, kako bi mogla da krene sa procesom šifrovanja fajla simetričnim RC4 ključem, kao i šifrovanja samog simetričnog RC4 ključa.

4.2.2 Proces komunikacije prilikom kreiranja RSA javnog ključa i privatnog ključa i njegovo slanje na e-mail

Sa slike 4-2.2. može se uočiti da Bob nakon prijema Alisinog e-maila sa zahtevom da joj dostavi svoj javni RSA ključ, staruje aplikaciju Project1.exe koja je u folderu SifrovanjeFajlova, koja kontaktira njegovu bazu ključeva kako bi proverio da li je već kreirao nekada javni i privatni RSA ključ za Alisu, i u slučaju da nije, kreira iste, automatski smešta svoj javni RSA ključ u ZaAlisaE-mailPublicKey.txt fajl u folderu SifrovanjeFajlova, automatski pokreće Microsoft Office Outlook, automatski kreira e-mail sa ovim fajlom u Attachment-u i definisanim naslovom i telom poruke i šalje takav e-mail na adresu Alise.

Proces koji obavlja hibridni sistem za šifrovanje-dešifrovanje kada Bob kreira svoj javni i privatni RSA ključ i šalje ga na e-mail Alise



Slika 4-2.2. Drugi proces koji se obavlja u hibridnom sistemu

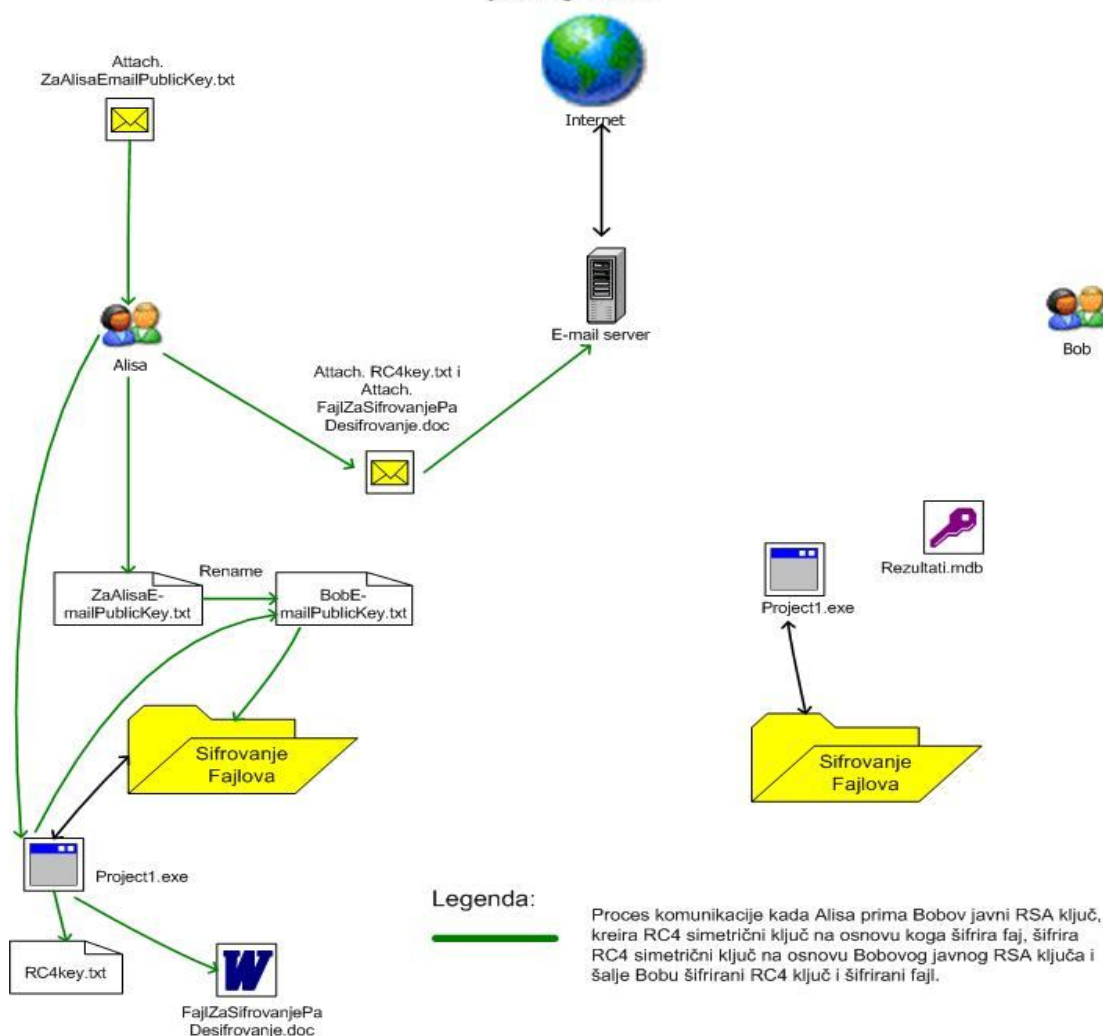
4.2.3 Proces komunikacije nakon prijema RSA javnog ključa, šifrovanja fajla i šifrovanja RC4 simetričnog ključa

Sledeća slika (4-2.3.) pokazuje proces u sistemu gde se vrši prijem e-maila koji sadrži fajl u Attachment-u sa Bobovim javnim RSA ključem, od strane Alise, nakon čega ona download-uje fajl (ZaAlisaEmailPublicKey.txt) iz Attachment-a u folder SifrovanjeFajlova, vrši promenu njegovog imena tako da ubuduće je asocira na onoga ko je kreirao svoj RSA javni ključ i smestio ga u fajl, dakle novo ime fajla bi trebalo da bude BobEmailPublicKey.txt.

Nakon ovoga, Alisa pokreće aplikaciju Project1.exe, kojom kreira svoj RC4 simetrični ključ, njime šifrira fajl sa imenom FajlZaSifrovanjePaDesifrovanje.doc koji može a ne mora biti u okviru foldera SifrovanjeFajlova, šifrira i automatski smešta svoj RC4 simetrični ključ u fajl RC4key.txt u folderu SifrovanjeFajlova.

Sledeće je da automatski pokreće Microsoft Office Outlook, automatski kreira e-mail sa fajlom (RC4key.txt) koji sadrži njen šifrovani RC4 simetrični ključ i šifrovani fajl (FajlZaSifrovanjePaDesifrovanje.doc) u Attachment-u i definisanim naslovom i telom poruke i šalje takav e-mail na adresu Boba.

Proces koji obavlja hibridni sistem za šifrovanje-dešifrovanje nakon prijema RSA javnog ključa

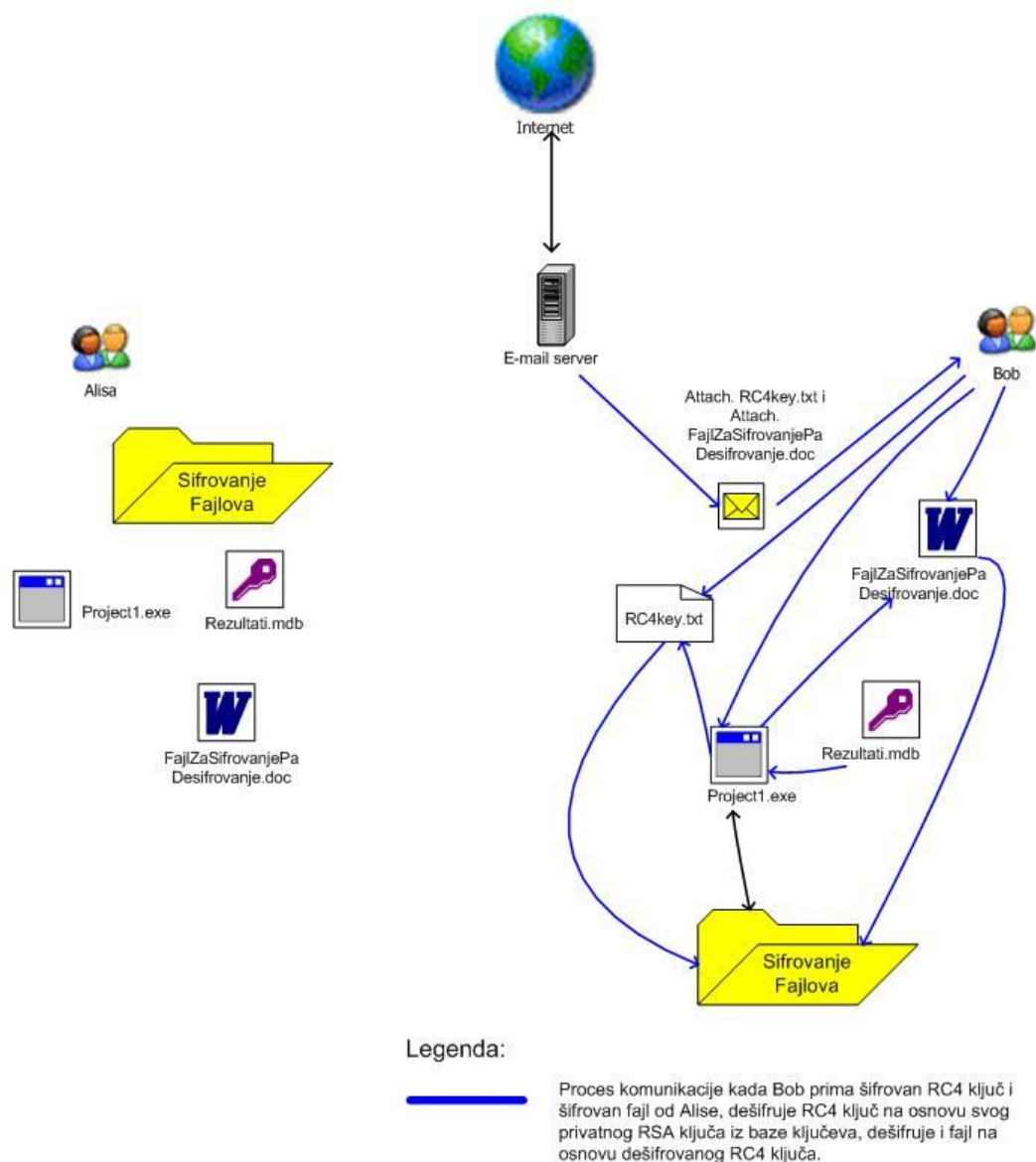


Slika 4-2.3. Treći proces koji se obavlja u hibridnom sistemu

4.2.4 Proces komunikacije prilikom prijema šifrovanog fajla i dešifrovanje istog

Nakon prijema e-maila kojeg je Alisa poslala Bobu sa šifrovanim RC4 simetričnim ključem smeštenim u fajl RC4key.txt i šifrovanim fajlom FajlZaSifrovanjePaDesifrovanje.doc, Bob ove fajlove download-uje, smešta u folder Sifrovanje Fajlova ili neki drugi, nakon čega startuje aplikaciju Project1.exe. Prvo se u aplikaciji kontaktira sa bazom ključeva Rezultati.mdb, odakle se dobija Bobov privatni RSA ključ kojim se dešifruje Alisin RC4 ključ smešten u fajl RC4key.txt, nakon čega Bob sa tako dešifrovanim RC4 simetričnim ključem, dešifruje i šifrovani fajl FajlZaSifrovanjePaDesifrovanje.doc. Završetkom ovog četvrtog procesa, završava se i kompletan rad hibridno sistema.

Proces koji obavlja hibridni sistem za dešifrovanje poslatog RC4 simetričnog ključa i fajla



Slika 4-2.3. Četvrti proces koji se obavlja u hibridnom sistemu

4.3 Neka rešenja problema koji su se javljali prilikom izrade ovog hibridnog sistema

Sam projekat izrade hibridnog sistema koji se i zove Project1, rađen je u VB6 IDE okruženju, i čine ga 6 formi, 6 modula i 2 modula klasa. Neka programerska rešenja formi, modula i modula klasa data su u pristupim radovima, ali ona koja su ključna, reklo bi se "noseća" u ovom projektu, su pre svega matematička rešenja problema opisanih u teoriji velikih brojeva i kongruenciji, a koja opet ne bi bila moguća bez rešavanja pretvaranja stringova cifara u heksadecimalni zapis, kome prethodi takozvano bitsko programiranje.

4.3.1 Programiranje matematike velikih brojeva

Ovo programiranje je na neki način, osnova, jer njime rešavamo sve druge probleme, pri čemu se ekstremno veliki prosti brojevi neophodni za generisanje RSA ključeva, svedu na dva niza sa ekstremno mnogo cifara i kao takvi nizovi, učestvuju u matematičkim operacijama sabiranja, oduzimanja i množenja, u posebnom VB6 modulu koji se zove LongMath.bas. U prethodnim pristupnim radovima je detaljno opisan ovaj modul.

Osnovna razlika ovog rada i prethodnih pristupnih radova, kada su prosti brojevi u pitanju, je u načinu generisanja prostih brojeva, gde se ovoga puta prosti brojevi biraju iz jedne Access baze, koja sadrži po desetak prostih brojeva za različite domene (deset prostih brojeva sa po 5 cifara, deset prostih brojeva sa po 10 cifara...deset prostih brojeva sa po 150 cifara). Ovo iz razloga skraćivanja vremena rada aplikacije, koja je ranije random funkcijom sama generisala proste brojeve sa zadanim brojem cifara. Sada je to dosta ubrzano, gde se random funkcija koristi samo prilikom izdvajanja broja iz baze po jedinom kriterijumu, a to je broj cifara tog broja.

Funkcije sabiranja, oduzimanja i množenja aktivno se pozivaju iz modula LongMath.bas u formu Kljucevi.frm, kako bi se kreirali RSA javni i privatni ključ, a detaljno su objašnjene u drugom pristupnom radu. Kôdni zapis modula i formi koje koriste ovo programiranje biće dat u prilogu ovoga rada.

4.3.2 Bitsko programiranje

Ovo programiranje je prisutno u nekoliko modula, a dosta je važno jer pretvara string sa ekstremno mnogo cifara, u bitski zapis, a onda takav zapis u heksadecimalni zapis, koji će kasnije da se koristi u modulu za neke matematičke operacije koje VB6 IDE može da izvodi samo sa ekstremnim brojevima datim u heksadecimalnom zapisu. Kôdni zapis modula i formi koje koriste ovo programiranje biće dat u prilogu ovoga rada.

4.3.3 Heksadecimalno programiranje

Nakon bitskog programiranja, trebalo je rešiti operaciju stepenovanje i operaciju moduo koje su neophodne za implementaciju RSA algoritma. VB6 IDE može raditi sa ovim operacijama, kada su u pitanju ekstremno veliki brojevi, samo ako se takvi brojevi pretvore u heksadecimalni zapis. U modulu ModExp.bas je dato programersko rešenje ovih

problema, a sam kôdni zapis biće dat u prilogu ovoga rada. U ovom modulu se vrše operacije stepenovanje i moduo nad ekstremno velikim heksadecimalnim brojevima.

Samo stepenovanje i mod funkciju je moguće brzo izvršiti samo ako su brojevi u heksadecimalnom zapisu. Koliko sporo i sa koliko malim brojevima, koji nisu u heksadecimalnom zapisu, može da radi RSA metoda, prikazano je u prvom pristupnom radu. Zato je bilo nužno predstaviti brojeve u heksadecimalnom zapisu i kao takve, implementirati ih u funkcije koje su bile dostupne na internetu i koje sam uz manje dorade, implementirao u projekat za rad sa RSA metodom, sa kojima je moguće izvršiti stepenovanje ekstremno velikih brojeva i ekstremno velikih eksponenata, kao i mod funkciju takvih brojeva.

Sa izvršenjem ovih operacija, i prethodno opisanih operacija nad unetim početnim varijablama u formi Kljucevi.frm, dobijaju se RSA javni i privatni ključ. Kôdni zapis modula i formi koje koriste ovo programiranje biće dat u prilogu ovoga rada.

4.3.4 Programiranje "mešanja" za RC4 algoritam

I ovaj problem se rešavao pretvaranjem simetričnog ključa u heksadecimalni zapis i njegovim "mešanjem" sa heksadecimalnim sadržajem svakog od 256 polja S kutije. Nakon svakog mešanja S kutije sa ključem, stvara se neko unutrašnje stanje, tako da nakon sumiranja svih stanja "mešanja", radi se XOR sa sledećim bajtom osnovnog teksta. Dakle, bilo je potrebno primeniti heksadecimalno programiranje na već dostupne kodne zapise za RC4 algoritam. Klasni modul clsRC4.cls radi ove operacije. Kôdni zapis modula i formi koje koriste ovo programiranje biće dat u prilogu ovoga rada.

5 Zaključak

Mnoge teorijske rasprave i praktični metodi šifrovanja podataka govore sledeće:

Simetrični šifarski sistem je oko hiljadu puta brži od asimetričnog, ali kod njega postoji problem prenosa ključa (ako se presretne ključ, podaci se mogu dešifrovati).

Kada se iskombinuju ova dva načina šifrovanja, dobija se željeni efekat: brzo šifrovanje sa sigurnim prenosom ključa. Ključ se, dakle, prenosi, ali šifrovan tako da ga samo osoba koja ima tajni ključ može dešifrovati.

Podaci se pre šifrovanja pakuju (vrši se kompresija), ako je moguće. Ovo je korisno iz dva razloga. Prvi je manja količina podataka za prenos. Drugi je dodatna sigurnost, jer se pakovanjem eliminiše pojavljivanje sličnih delova u izvornoj datoteci. Mnoge tehnike kriptanalize iskorišćavaju baš te slične delove da bi probile zaštitu. Naravno, fajlovi koji su ili prekratki za pakovanje ili se ne mogu spakovati dovoljno, ostavljaju se u izvornom obliku. U ovom radu se izbeglo pakovanje podataka, jer se radi o probnoj verziji, koja bi trebalo da zadovolji najminimalnije zahteve kada je šifrovanje i zaštita od kriptanalize u pitanju, a da opet pokaže praktično sprovođenje teorijskih tvrdnji o efikasnosti i efektivnosti hibridnog sistema.

Posle pakovanja, pravi se privremeni ključ, odnosno slučajan broj. Ovaj ključ ima jednokratnu upotrebu, jer se koristi da bi se podaci šifrovali simetričnom enkripcijom. Zatim se šifrue samo privremeni ključ asimetričnom enkripcijom i pridružuje šifrovanim podacima.

Dešifrovanje se vrši suprotnim procesom. Prvo se pomoću tajnog ključa dešifrue privremeni ključ, a njim se onda dalje dešifruju podaci.

Ovaj rad se oslanja na sve gore navedene principe. Njegova suština je u praktičnoj realizaciji aplikacije Project1.exe. Ova aplikacija, za razliku od komercijalnih, funkcioniše i za realno izabrane velike dužine ključeva.

Moglo bi se zaključiti, na osnovu rada same aplikacije, kao i prikupljanja materijala za izradu rada, da on upravo i potvrđuje navedene tvrdnje, mada bi se pravi zaključak mogao doneti tek nakon njegove stalne eksploatacije i to od strane više korisnika.

U prvom i u drugom pristupnom radu se težilo što većoj praktičnosti aplikacije, dakle što brži rad sa što većim ključevima, što se i postiglo u ovom završnom radu, a ostalo je da se eventualno u budućnosti više pažnje pokloni dizajnu aplikacije, kao i ispravci eventualnih nedostataka, koji će se javiti u stalnoj eksploataciji aplikacije.

6 Literatura i Internet linkovi

1. William Stallings– **Cryptography and Network Security Principles and Practices, Fourth Edition**, Prentice Hall, November 2005
2. Alfred J. Menezes- **Applied Cryptography, Fifth Printing**, CRC Press, October 1996
3. <http://www.di-mgt.com.au3>, RSA Algorithm
4. <http://www.123eng.com/sourcecode>, **RSA Algorithm (Mini Project)**
5. <http://rsa.googlecode.com/svn-history/r55/rsa/trunk/docs/rsa.pdf>
6. <http://pajhome.org.uk/crypt/rsa/rsa.pdf>, RSA Algorithm
7. <http://en.wikipedia.org/wiki/RSA>, RSA
8. http://www.experts-exchange.com/Programming/Languages/.NET/Q_22986420.html, Convert large Hex to Decimal in VB.NET
9. <http://www.hexprobe.com/index.htm>, Hex converter
10. <http://en.wikipedia.org/wiki/RC4>
11. <http://www.vocal.com/cryptography/rc4.html>
12. <http://www.tech-faq.com/rc4.shtml>